**Міністерство освіти і науки, молоді та спорту України**

ЖИТОМИРСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

**Методичні вказівки**

До лабораторних робіт з дисципліни «Програмні оболонки і пакети»

Частина ІV: MS Access

для студентів напрямку «Програмна інженерія»

Затверджено

на засіданні кафедри

програмного забезпечення систем

Протокол №8

від 30.06.2015

Житомир, ЖДТУ

2015р.

Автори: Н.Білодід, О.Власенко, І.Оринчак.

Кафедра програмного забезпечення та обчислювальних технологій

ЗМІСТ

Вступ 3

Лабораторна робота № 2.1 MS Access - СУБД в складі MS Office. 4

Лабораторна робота № 2.2 MS Access - работа з формами 23

Лабораторна робота № 2.3 MS Access - запити і створення складних форм. 36

Лабораторна робота № 2.4 MS Access - Звіти, Кнопкові форми та параметри запуску БД 49

Контрольні питання: 59

Література 60

# Вступ

Метою дисципліни є вивчення теоретичних основ і принципів побудови сучасних прикладних програм (електронні табличні процесори, системи підготовки текстів, бази даних та СУБД тощо).

Предмет навчальної дисципліни — система засобів автоматизації обробки та використання різноманітної інформації.

У результаті вивчення дисципліни студенти повинні:

* знати теоретичні основи інформатики та експлуатаційні характеристики комп’ютерної техніки;
* бути ознайомленими із системами обробки інформації, з методами запровадження діалогу в процесі виконання конкретних завдань;
* вміти користуватися можливостями програмного забезпечення для реалізації прикладних завдань, що розраховані на конкретного споживача;

Програма курсу передбачає навчання у формі лекцій, практичних і лабораторних занять. Для практичного засвоєння основних тем дисципліни передбачені лабораторні заняття із застосуванням ПК у класах комп’ютерного центру університету. Саме ця частина і висвітлена в методичних вказівках, які ви зараз читаєте.

Вивчення дисципліни дає підґрунтя для подальшого засвоєння можливостей використання комп’ютерної техніки у спеціальних дисциплінах навчального плану студентів економічних спеціальностей усіх форм навчання.

# Лабораторна робота № 2.1 MS Access - СУБД в складі MS Office.

Мета роботи: Напрацювання перших навичок роботи з MS Access. Навчитись створювати оригінальні БД для власних потреб.

### Короткі відомості

Бази даних (БД) - це один із напрямків у розвитку прикладного програмного забезпечення. БД призначені для зберігання і обробки великих обсягів інформації.

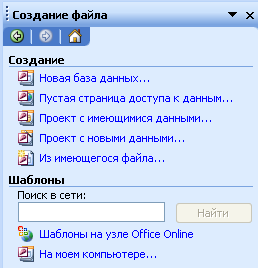
CУБД - система управління базами даних.

В основу побудови СУБД MS Access покладено реляційна модель БД. Реляційна база даних - база даних, заснована на реляційної моделі даних. Слово «реляційний» походить від англ. relation (відношення). Використання реляційних баз даних було запропоновано доктором Коддом з компанії IBM в 1970 році.

Пакет MS Access поєднує в собі такі (зовні суперечливі) якості, як простота використання і практично необмежені можливості маніпулювання даними. Ви можете, нічого не знаючи про програмування і про теорію БД, створити за допомогою майстрів MS Access повнофункціональну БД для своїх потреб. А якщо Ви кваліфікований програміст і знайомі з програмуванням на VBA, то Ви можете створювати додатки будь-якої складності і для будь-яких потреб, Access надає для цього всі можливості. MS Access входить до складу лише MS Office Professional.

#### Створення бази даних.

Після запуску MS Access в області задач головного вікна з’являється задача «Создание файла», де користувачеві надається можливість вибору однієї із наступних дій:



Вибираючи «Новая база данных», ми викликаємо діалогове вікно «Файл новой базы данных», в якому називаємо базу (за замовчуванням db1) вибираємо місце зберігання (за умовчанням «Мої документи»).

Після збереження нової бази даних, в робочому вікні MS Access з'являється вікно БД, де користувач може виконувати всі дії з базою.

Робочий файл MS Access являє собою файл-контейнер з розширенням **\*.accdb**, він містить в собі весь набір компонентів СУБД. Копіювати, видаляти або переміщати БД можливо тільки в такому контейнері.

**\*.accdb-файли** призначені для створення і редагування баз даних. Тобто в таких файлах активними є всі типи конструкторів: таблиць; запитів; форм; звітів; редактора Visual Basic for Application. Цей тип файлів є основним для MS Access.

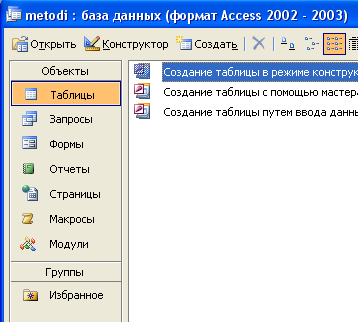
Крім файлів \*.mdb, MS Access працює також з файлами \*.mdz та \*.mde.

**\*.accdz-файли**, містять «Шаблон майстра бази даних». За допомогою майстрів можна швидко створити кілька типів баз, покрокові відповідаючи на питання діалогу. Наприклад, вибираючи шаблон Склад, результатом роботи файлу Склад. **acc**dz стане файл готової робочої бази Склад1. **acc**db.

**\*.accde-файли** призначені тільки для роботи з готовою і повністю налагодженою базою. \*.**acc**de-файли створюються з \*.**acc**db-файлів шляхом їх компіляції. В таких файлах вже не можна змінити форми, звіти, запити, модулі. Можна тільки вносити дані, змінювати їх, друкувати заготовлені звіти. Цей тип файлів зручний в тому випадку, коли розробник бази передає її в експлуатацію користувачам. Варто пам’ятати, що основний \*.**acc**db-файл необхідно зберігати для можливості внесення змін в базу, коли виникне така необхідність. Перетворення із **acc**db в **acc**de виконується через Файл – Сохранить и опубликовать – Создать **acc**de-файл.

#### Головне вікно MS Access

Більшість елементів вікна MS Access є стандартними для Windows об'єктами: заголовок вікна, рядок меню, панель інструментів. Разом з тим, MS Access має ряд особливостей. Перша – це вікно бази даних.



Це вікно відображає всі компоненти БД, та надає можливість з ними розпочати роботу.

Ще одна особливість полягає в тому, що панелі інструментів тут постійно змінюються. На екран виводиться та панель, яка відповідає виділеному об'єкту і його режиму роботи.

Таблиці– в реляційних базах вся інформація зберігається у вигляді таблиць. Кожен рядок таблиці являє собою запис, а стовпець - поле. Запис містить набір даних про один об'єкт, а поле - однорідні дані про всі об'єкти.

Запити – виконують широке коло функцій. Головна функція - відбір даних на підставі критеріїв, що дозволяє вибрати з бази даних необхідну інформацію.

Форми – представляють собою бланк, який підлягає заповненню. Форма дозволяє спростити процес заповнення бази. Також форми містять активні об'єкти з елементами програмування.

Звіти – служать для відображення даних з таблиць, що відповідають певним критеріям відбору даних. Також існують запити для виконання певних дій з таблицями і даними.

Макроси – дозволяють автоматизувати процес роботи з базою: заповнення, відбору інформації і т.д.

Модулі – процедури обробки подій або виконання обчислень, написаних на мові програмування Visual Basic for Application (VBA). За кожним активним елементом форми закріплена відпоідна процедура, що містить код, який виконується при активізації такого елементу. Саме модулі дають програмісту можливість перетворити MS Access в повноцінній програмний засіб.

#### Таблиці

Таблиці – основний об'єкт БД. На основі таблиць створюють форми, запити і звіти. В реляційних базах таблиці містять опис всіх даних, які зберігаються в них. Оскільки кожне поле таблиці зберігає однорідні дані, то саме поле і зберігає опис таких даних. Інакше це називається – тип поля. Типи полів в MS Access наступні:

Текстовий – містить текст, обсяг якого не повинен перевищувати 255 символів (довжина поля регулюється за допомогою властивості Розмір поля)

Поле MEMO – зберігає текстові або текстові і числові дані об'ємом не більше 64 000 символів (поля цього типу не індексуються)

Числовий – містить числові значення, діапазон яких визначається властивістю Розмір поля.

Дата / час – містить значення дати і часу

Грошовий – зберігаються числа, що містить 15 розрядів ліворуч від десяткової точки і 4 праворуч від неї. Пов'язаний з модулем «Языки и стандарти».

Лічильник – містить число, яке автоматично збільшується на одиницю при додаванні в таблицю нового запису.

Логічний – зберігають значення Так або Ні, не індексуються.

Поле об'єкту OLE – містять об'єкти, оброблені OLE‑сервером, не індексуються, розмір не повинен перевищувати 128 Мб.

Гіперпосилання – в полях цього типу зберігаються гіперпосилання, які представляють собою шлях до файлу або адресу в мережах Internet або Intranet.

Окрім типу, кожне поле має набір додаткових параметрів, що уточнюють характеристики даних, та/або задають додаткові можливості при введенні даних в таблицю. Називаються такі додаткові параметри «свойства».

Размер поля - визначає Max довжину текстового або числового поля (при дуже великому розмірі неекономно витрачається пам'ять, а при малому спотворюється вміст полів). Розмір текстового поля 1-255 символів.

Розмір поля числових полів:

|  |  |
| --- | --- |
| Байт | Целые числа от0 до255 (поле занимает 1 байт) |
| Целое | Целые числа от -32768 до 32768 (поле занимает 2 байта) |
| Длинное целое | Целые числа от -2147483648 до 2147483648 (поле занимает 4 байта) |
| С плавающей точкой (4 байта) | Дробные числа с шестью знаками после запятой от -3,402823Е38до 3,402823Е38) |
| С плавающей точкой (8 байтов) | Дробные числа с десятью знаками после запятой  От –1.79769313486231E308 дo1.79769313486231E308 |

За замовчуванням розмір текстового поля становить 50 символів, а розмір числового - Довге ціле.

Формат поля - визначає спосіб відображення вмісту на екрані і на папері після друку. Число десяткових знаків - визначає кількість розрядів в дробової частини числа.

Маска ввода - застосовується для введення дати і тексту

Маска введення дозволяє створити в полях введення потрібні для Вас умови. Тобто Ви можете змусити користувача вводити тільки ту інформацію, яка вам потрібна. Наприклад, потрібно вводити телефонний номер з урахуванням коду міста. Тоді створіть маску:

(0000) 00-00-00. У полі введення отримаємо: (\_\_\_\_) \_\_-\_\_-\_\_

Маски можуть містити будь-які символи, які будуть стояти в полі введення незалежно від бажання користувача (у нашому прикладі - це дужки і знаки «-») та службові символи, які визначають місця, для введеної користувачем інформації (у прикладі - це 0, означає, що користувач обов'язково повинен ввести на це місце, будь яку цифру).

Подпись - містіть напис, який НЕ збігається з ім'ям поля. Це зроблено для більш зрозумілого відображення вмісту поля в режимі таблиці. Також підпис, якщо він існує, виводиться поряд з полем у формі чи звіті замість фактичної назви поля.

Значение по умолчанию – це значення, яке буде автоматично вводитись в поле, для кожного нового запису в таблиці. Наприклад, функція Date() вставить в поле поточну дату.

Условие на значение - умова, на яку буде перевірятись значення поля, при заповненні таблиці.

Сообщение об ошибке – визначає текст повідомлення, яке з'явиться на екрані, при введенні недопустимого значення.

Обязательное поле – ця властивість поля визначає, чи потрібно поле заповнювати обов’язково, чи можна залишити пустим.

Индексированное поле – визначає чи встановлено для поля індекс. Індекс використовується для прискорення пошуку та сортування по даному полю. Якщо поле індексоване, то потрібно вказати додатково: дозволяються чи ні в такому полі однакові значення.

Майстер підстановок. MS Access містить механізм, реалізований в Конструкторі таблиці, що дозволяє підставляти дані з інших таблиць, відповідаючи на питання майстра. Додаткові властивості такого поля представлені на вкладці «Підстановка» вікна конструктора таблиць.

#### Зв’язки між таблицями.

В реляційних базах даних міжтабличні зв'язки дозволяють уникати зберігання надлишкових даних, за рахунок збереження в таблицях тільки унікальних даних. А ті дані які повторюються – переносяться в окрему таблицю.

Уявімо собі перелік мінімально необхідної інформації для покупки товарів у різних фірм:

Найменування товару

Кількість закупленого товару

Ціна покупки за одиницю

*Назва фірма-постачальника товару*

*номер телефону фірми, адреса і поштовий індекс.*

*Керівник*

*Коротка характеристика*.

Якщо записувати всі ці відомості в таблицю «ПОКУПКИ», то, наприклад, телефон фірми та адресу потрібно буде повторювати кожен раз, коли ми заносимо новий запис. У той же час, і так зрозуміло, що запис про товар, закупленому в однієї і тієї ж фірми, повинен містити один і той же телефон і адресу. Те ж стосується і назви фірми та керівника і короткої характеристики. Тобто, пункти 1‑3, наведеного переліку даних, є унікальними для процедури закупівлі, а пункти 4‑7 є такі що можуть повторюватись.

Більш ефективним є збереження відомостей про фірми-постачальники в окремій таблиці «ФИРМЫ». Потім слід в таблиці «ПОКУПКИ» просто встановити вказівник на запис в таблиці «ФИРМЫ», що містить потрібні відомості.

### Робоче завдання:

1. Запуск MS Access, створення нової БД, знайомство з елементами його робочого вікна.

Створення таблиць для побудови БД торгівельного підприємства. Знайомство з типами даних, що зберігаються в таблицях, створення ключових полів в таблицях.

Робота з Майстром підстановок в Конструкторі таблиці.

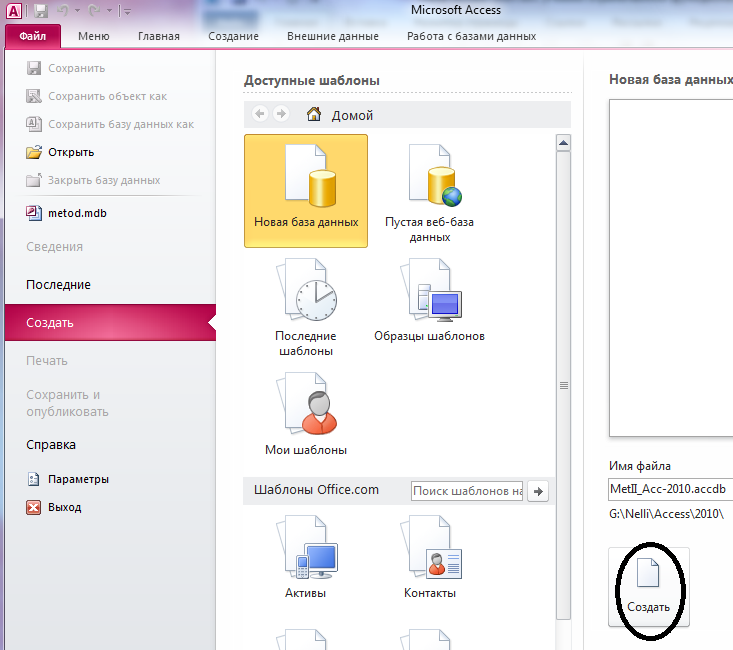
Створення зв’язків між таблицями.

Заповнення таблиць даними.

Збереження БД. Процедура стиснення та відновлення файлу БД.

### Порядок виконання завдання

1. Запустити MS Access, використавши головне меню Windows: «Пуск» - «Все программы» - «Microsoft Office» - «Microsoft Office ***MS Access***». Ознайомитись з вікном MS Access, елементами управління. Створіть нову БД:

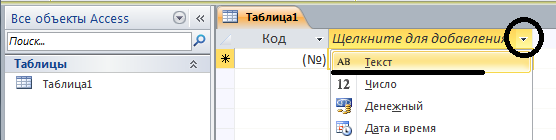


2. Створення таблиць.

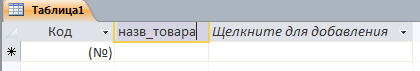
Ассеss відкриє порожню таблицю з ім'ям «Таблиця1», при цьому курсор буде знаходитися в стовпці «Щелкните для добавления». Зліва від вікна з таблицею розташовується область переходів. У ній відображаються всі об'єкти бази даних, розділені на категорії. Поки що тут тільки одна таблиця.

Першим полем в новій таблиці є поле «Код», це поле було створено автоматично. У нього тип даних – «Счетчик». Таке поле є числовим і унікальним для кожного запису, воно буде автоматично послідовно нарощуватись для кожного нового запису.

Для створення нового поля (в якому ми будемо зберігати назви товарів), натисніть кнопку, яка відкриває список. Це список типів даних, які можуть міститись в новому полі. Виберіть тип «Текст».



Буде створено нове поле з іменем «Поле1». Замість цієї назви введіть іншу: «назв\_товара».



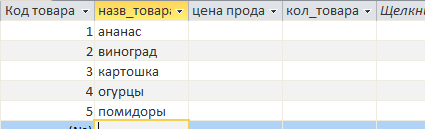
Таким чином створюється структура таблиці в режимі таблиці.

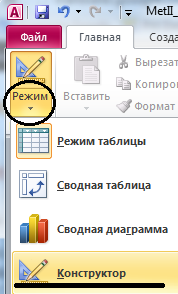
Структура - це назви полів в таблиці і типи даних, що зберігаються в цих полях.

Створіть самостійно ще декілька полів в таблиці: тип «Денежный», назва: «цена продажи» та тип: «Число» назва: «кол\_товара».

На полі «Код» клацніть правою кнопкою миші і виберіть в контекстному меню команду «Переименовать поле». Змініть назву поля на «Код товара». Ця вправа показує, яким чином можна змінювати існуючі поля.

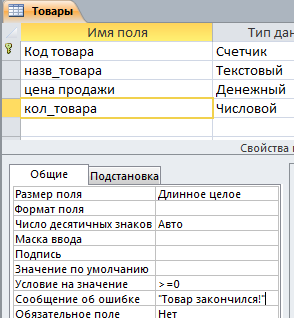
Тепер можна заповнювати таблицю значеннями. Введіть назви товарів (краще використати назви із попередніх лабораторних робіт), іншу інформацію введемо пізніше. Приклад наведено нижче:



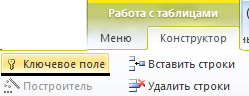
Створену таблицю треба зберегти. Для цього натисніть кнопку "Сохранить" на панелі швидкого доступу і дайте ім'я таблиці: «Товары».

Всю роботу, що ми з вами виконали, ми робили в режимі таблиці. Але це – спрощений варіант побудови таблиць. Для розширених можливостей конструювання таблиць слід використовувати режим конструктору. Переведіть нашу таблицю в цей режим, так як це показано на малюнку. Відкриється вікно таблиці у режимі конструктора.

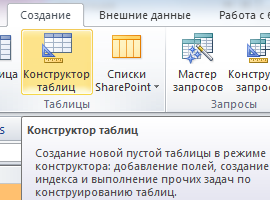
Вікно складається з двох частин. У верхній частині відображається список полів таблиці, а в нижній - властивості виділеного поля



Для поля кол\_товара використовуємо додаткові сервісні характеристики: Условие на значение встановіть значення >=0 (більше або дорівнює нулю), Сообщение об ошибке введіть текст «Товар закончился!». Такі установки не дозволять вводити в поле негативні числа, при цьому на екран буде виводитись відповідне повідомлення.

Зверніть увагу, що поле «Код товара» має значок ключа. Таке поле називається ключовим, воно автоматично заповнюється деякими числами. Ці числа не обов'язково утворюють безперервну послідовність. Єдиною умовою є їх унікальність. Значення ключового поля відрізняються для всіх записів таблиці, що дозволяє однозначно ідентифікувати запис. Такі поля використовуються для формування зв'язків між таблицями. При створенні наступних таблиць Вам потрібно самостійно створювати ключові поля. Для цього використовуйте кнопку:

Тепер самостійно за допомогою конструктору



створіть наступні таблиці:

|  |  |
| --- | --- |
| Фирмы | |
| Поле | Тип данных |
| Код фирмы | Счетчик (ключ) |
| название | текст |
| адрес | текст |
| руководитель | текст |

|  |  |
| --- | --- |
| Покупки | |
| Поле | Тип данных |
| Код покупки | Счетчик (ключ) |
| Код фирмы | числовой (размер: длинное целое) |
| Код товара | числовой (размер: длинное целое) |
| цена покупки | денежный |
| количество | числовой (размер: целое) |
| дата покупки | дата/время |

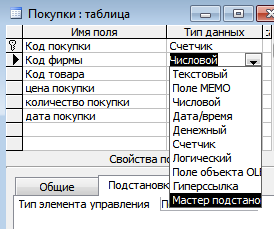
|  |  |
| --- | --- |
| Продажи | |
| Поле | Тип данных |
| Код продажи | Счетчик (ключ) |
| Код товара | числовой (размер: длинное целое) |
| количество | числовой (размер: целое) |
| дата продажи | дата/время |

Заповніть таблиці Фирмы та Товары записами, використавши данні з попередніх лабораторних робіт.

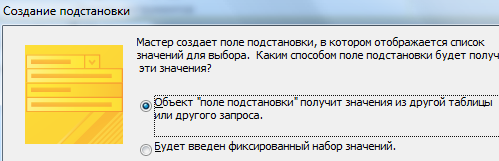
3. Майстер підстановок. В таблицях Продажи і Покупки відсутні назви фірм і товарів, замість них використано поля Код товара та Код фирмы.

Зараз увага – саме так треба створювати реляційні бази!

Але користувач бази повинен бачити не коди а назви! Для цього відповідні поля повинні **автоматично** заповнюватись кодами із відповідних таблиць, а користувач бачити зрозумілі йому назви. Для такої організації роботи використовується *механізм підстановки*.

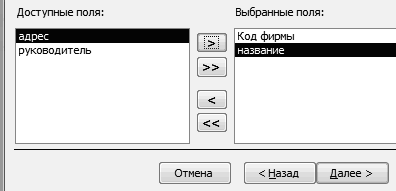


Після вибору типу даних «Мастер подстановки», запускається майстер. На першому кроці потрібно вибрати один із запропонованих варіантів створення підстановки. В нашому випадку обираємо перший варіант.



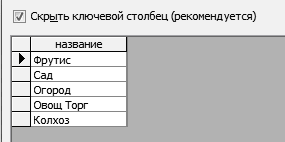
На другому кроці майстра просто вкажіть таблицю Фирмы

Третій крок визначає поля вибраної таблиці, з яких буде братись інформація для підстановки. Тут потрібно вибрати поля Код фирмы та название.



Четвертий крок дозволяє втановити сортування списку, ми його пропускаємо.

П’ятий крок показує, як буде виглядати список і пропонує сховати ключове поле. В нашому випадку – ключове поле Код фирмы Це якраз те що нам і порібно: бачити назву, а запам’ятовувати код!

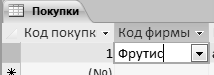
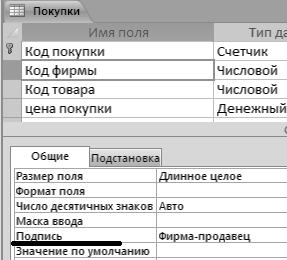


Останній крок майстра пропонує надати ім’я поля. Залиште текст «Код фирмы» Тиснік кнопку «Готово» - робота майстра закінчена!

Після завершення роботи майстра автоматично створюється зв’язок між таблицями Покупки и Фирмы. На екрані з’явиться запитання про збереження структури таблиці – підтвердіть цю дію.

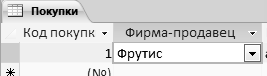
Зверніть увагу!

Тепер поле Код фирмы показує назву фірми, а підпис поля в режимі таблиці відображає назву поля, і це може стати незручністю для користувача.

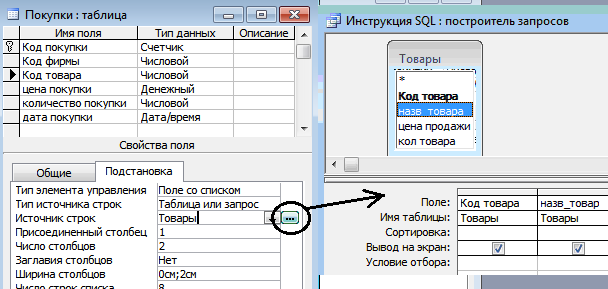


Для виправлення такої ситуації потрібно скористатись властивістю поля Подпись:

Запишіть сюди текст, який буде відображати суть самої підстановки і буде зрозумілий користувачеві – «Фирма-продавец». Тепер в режимі таблиці ми побачимо:



Створити механізм підстановки можна і вручну. Для цього потрібно використати закладку Подстановка в Конструкторі таблиці для відповідного поля. На малюнку нижче наведено приклад, як це можна зробити:



В першому рядку встановіть значення Поле со списком. В Другому – залиште без змін. В третьому спочатку виберіть із списку таблицю Товары, потім натисніть кнопку, обведену на малюнку. На екрані з’явиться конструктор запиту. Виберіть поля Код товара та назв\_товара. Закрийте конструктор, Access видасть запитання – підтвердіть внесення змін у таблицю.

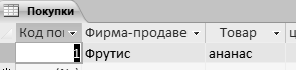
Тепер основна хитрість ручного створення механізму підстановки:

1) треба встановити параметр Число столбцов = 2

2) параметр Ширина столбцов: треба вказати два значення 0см та 2см, розділені знаком крапка з комою.

Саме таким чином ми ховаємо перше ключове поле і виводимо тільки зрозумілий для користувача текст – назви товарів.

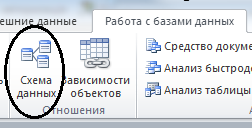
На завершення, змініть для цього поля властивість Подпись (так, як це було роз’яснено вище) – вкажіть текст підпису «Товар». Ви повинні отримати такий вигляд в режимі таблиці:



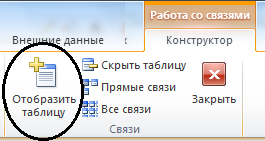
4. Створення зв’язків між таблицями інколи відбувається автоматично, наприклад, при роботі майстрів. Так в нашій роботі, при використанні Майстра підстановки між таблицями Покупки та Фирмы буде створено зв’язок по полю код фирмы.

В наступній вправі вам потрібно створити зв’язок між таблицями Продажи і Товары.

Для цього на закладці «Работа с базами данных» натисніть кнопку «Схема данных»

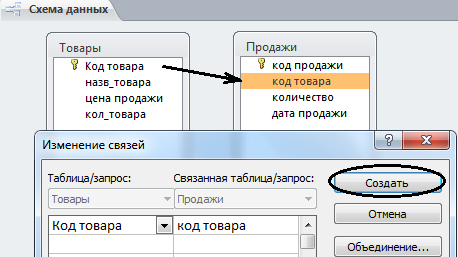


З’явиться спеціальне вікно, яке відображає всі існуючі зв’язки між таблицями и запитами в поточній БД. Натисніть кнопку «Отобразить таблицу»

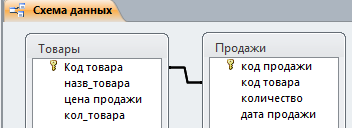


У вікні «Добавление таблицы» виберіть таблицю Продажи і натисніть кнопку Добавить. Аналогічно добавте таблицю Товары.

Візьміть мишкою поле Код товара із таблиці Товары і перетягніть на поле код товара із таблиці Продажи. З’явиться вікно «Изменение связей»



Після натискання кнопки «Создать» отримаємо наступну картинку у вікні «Схема даних»:

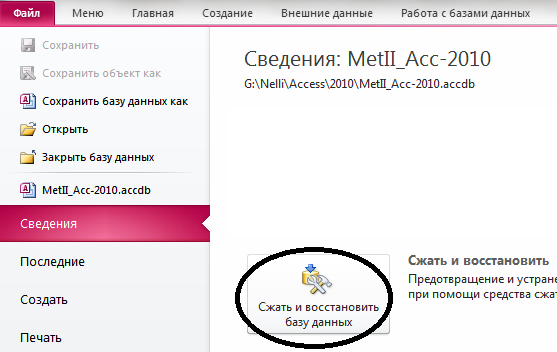


5. Заповнення таблиць даними. На даному єтапі роботи навчіться заповнювати таблиці даними. В кожну із створених таблиць введіть дані. Можете використати інформацію із попередніх лабораторних робіт, або придумати нову.

6. Збереження БД відбувається кожного разу, коли зберігаються окремі її компоненти.

Пригадаємо: робочий файл MS Access являє собою файл-контейнер з розширенням **\*.accdb**, він містить в собі весь набір компонентів СУБД. Такий тип файлів має свою специфіку. В межах такого файлу відбувається вся робота з базою. В тому числі, створення тимчасових таблиць, розгортання майстрів. Такі дії приводять до виділення у файлі додаткового місця для тимчасових об’єктів, яке потім очищується і залишається не використовуваним. Саме для оптимізації розміру файлу, а також для виправлення можливих внутрішніх помилок, передбачено спеціальний режим: *стиснення та відновлення файлу*.

Ваше завдання на даному етапі роботи – виконати цю процедуру. Для цього закрийте всі об’єкти бази, попереднє зберігши їх, та перейдіть на вкладку Файл – Сведения - Сжать и восстановить базу данных.



# Лабораторна робота № 2.2 MS Access - робота з формами

Мета роботи: ознайомитись з функціями об’єктів форми; навчитись створювати форми за допомогою функції автоформи, майстра форм, а також в режимі конструктора; налаштовувати зовнішній вигляд форми з урахуванням уподобань користувачів та особливостей даних.

### Короткі відомості

Форму можна розглядати в якості діалогового вікна, через яке користувачі можуть переглядати вміст БД та змінювати інформацію в ній. Ефективна форма прискорює використання інформації, що міститься в базі даних, оскільки користувачам не потрібно буде виконувати пошук необхідних їм відомостей. Візуальна привабливість форми робить роботу з нею більш приємною і ефективною.

При створенні форми необхідно завжди мати на увазі наступні два моменти. Використання форм буде виправданим, якщо з її допомогою забезпечується ясність представлення даних і можливість їх контролю.

Інформація баз даних зберігається в таблицях. Часто ці таблиці є дуже великими. Можна, звичайно, вводити дані відразу ж в таблицю або ж зчитувати їх з неї. Однак цей процес, як правило, виявляється дуже незручним. Чим більша таблиця, тим складніше гарантувати, що дані заносяться в потрібне поле (стовпець) і запис бази даних (рядок).

Використання форми допомагає користувачеві виконувати тільки ті дії, які необхідні для введення або перегляду потрібних йому даних. Розробка форми повинна виконуватися таким чином, щоб можна було легко зрозуміти її призначення і прочитати представлену в ній інформацію. У форму можна включити написи з описом даних, інструкції щодо введення інформації, розташовуючи їх поруч з відповідними робочими областями.

Форми дозволяють вибирати з безлічі полів таблиці саме ті поля, до яких потрібно привернути увагу користувача. Ця можливість означає також, що при бажанні можна заблокувати доступ до решти даних таблиці. Наприклад, людина, що вводить прізвища в список співробітників, не зможе переглядати відомості про зарплату.

#### Типи форм

Є декілька типів форм. Найбільш поширені два типи:

**Форма у стовпець** - відображає одночасно тільки один запис, кожне поле якої наводиться на окремому рядку і має відповідний підпис.

**Стрічкова форма** - відображає одночасно всі записи, при цьому для кожного запису відображається кожне поле, а назви полів відображаються вгорі форми.

Додатково можна створювати:

**Підпорядкована форма**: виглядає як частина іншої форми (головної). Фактично ж, коли головна форма відображає дані з однієї із таблиць/запитів, то підпорядкована форма - з іншої, зв’язаної з головною таблицею. Користувач може вводити дані в одну форму і, одночасно, переглядати/редагувати пов'язані з нею дані з іншої таблиці. Підпорядкована форма завжди виводиться у режимі таблиці.

**Форма-повідомлення**: у такій формі можуть відображатися тексти для користувача. У ній можуть наводитися інструкції з використання бази даних або усунення проблем, що виникають при її використанні. Така форма не базується на таблицях чи запитах.

**Кнопкова форма**: форма, що відображається у вигляді вікна із кнопками. Натискання на кнопки призведуть до певних дій. Кнопкова форма забезпечує зрозумілий і зручний спосіб для відкриття форм, звітів та інших об'єктів. Це допомагає користувачам виконувати саме ті дії, які від них потрібні, а також приховувати від них ті частини бази даних, до яких вони не повинні мати доступу. Кнопкову форму можна створити самостійно, або за допомогою спеціального майстра.

#### Методи створення форм

В Access є три методи створення форми: за допомогою функції автоформи, майстра форм, а також в режимі конструктора.

**Автоформи** є найбільш простим способом створення форми. Коли нова форма буде містити всі поля таблиці або запиту і користувач не хоче займатися визначенням її зовнішнього вигляду, автоформа є найкращим вибором.

При використанні автоформи користувачеві не задаються будь які питання і не пропонуються варіанти вибору. Робота функції автоформи виконується за два кроки: виберіть потрібний тип автоформи, виберіть потрібну таблицю або запит. Все інше зробить за вас Access.

**Майстер форм** задає користувачеві питання, відповіді на які визначають вид створюваної ним форми. Використання майстра форм найбільш ефективно, коли користувач хоче послідовно пройти по всіх кроків створення форми.

Майстру потрібно повідомити такі відомості:

* З якої таблиці або запиту беруться дані, які використовуються у формі.
* Які поля будуть використовуватися у формі.
* Який макет форми потрібно застосувати.
* Який візуальний стиль потрібно застосувати.

Така робота займе трохи більше часу, ніж функція автоформи, але і результат буде більш наближеним до вимог користувача.

**Конструктор форм**. Метод створення форми в режимі конструктора є найбільш ефективним, коли користувач хотів би при цьому в максимальній мірі реалізувати свої плани та уподобання. У режимі конструктора отриманий результат повністю залежить від розробника.

Робота з формою в режимі конструктора ведеться за допомогою елементів форми, які вибираються на панелі інструментів «Элементами управления». Це елементи, такі як поля, прапорці, лінії, малюнки, написи, кнопки. Їм можна задати властивості і форматувати у відповідності зі своїми смаками та уподобаннями.

У конструкторі можна переміщати елементи управління з місця на місце. Наприклад, можна згрупувати в одному місці поля адреси (вулиця і номер будинку, місто, область, поштовий індекс), однак при цьому потрібно вирішити, на скількох рядках вони будуть розташовуватися - двох, трьох або чотирьох? Який з варіантів є найкращим, повинен визначити творець форми.

#### Режими форм

Будь яка форма може знаходитись в одному з таких режимів:

* Режим конструктора
* Режим макета
* Режим форми

**Режим конструктора** призначений для редагування форми. Так, наприклад, форму можна створити за допомогою функції авто форми, або майстра, а потім відредагувати в режимі конструктора.

**Режим макета** – це другий із режимів редагування форми. Його особливістю є можливість змінювати розмір і місцеположення елементів форми. В цьому режимі не можна заповнювати форму даними, але самі дані відображаються (на відміну від режиму конструктора, в ньому не видно як будуть виглядати дані на екрані).

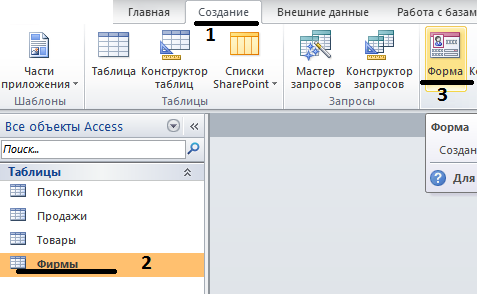
**Режим форми** – це один із робочих режимів форми. Щоб подивитися, як буде виглядати форма, коли користувач приступить до роботи з нею, її слід відкрити в режимі форми, і, перемикаючись між режимом форм і режиму конструктора, вносити в неї потрібні зміни, домагаючись бажаного результату. Особливістю цього режиму є відображення на формі всіх елементів управління і оформлення форми: кнопки, надписи, лінії, фоновий малюнок.

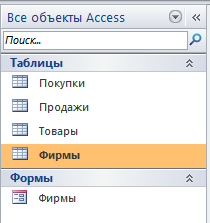
### Робоче завдання:

1. Створити форму Фирмы в автоматичному режимі – одним натисканням кнопки.
2. Створення форм за допомогою майстра.
3. Редагування форм в режимі конструктора:
   1. Прибрати зайві елементи у формі Фирмы та заборонити доступ до ключового поля.
   2. Додати у форму Товары можливість відображення товарів у малюнках.
   3. Створення активних елементів на формі – кнопки для виконання дії: «закрити форму».
   4. Редагування форми Продажи.
4. Створення кнопкової форми для запуску різних об’єктів БД.

### Порядок виконання завдання

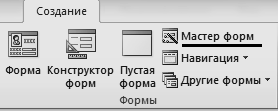
1. Для автоматичного створення форми використовується кнопка Форма на закладці «Создание» група «Формы». Для для того, щоб кнопка активізувалась спочатку виділіть таблицю Фирмы.



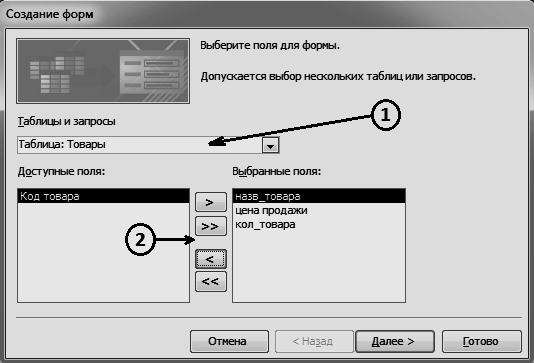
Це – найпростіший метод створення форми в 1 натискання мишею. Залишимо поки що цю форму такою, яка вона є. Для завершення процесу, Вам потрібно закрити форму, при цьому Access видасть запит на збереження нової форми. Підтвердіть цю дію.

Тепер у вікні БД, на закладці Формы з’явиться новий об’єкт:

2. Переходимо до роботи з майстром створення форми. Будемо створювати форму на основі таблиці Товары. Тому, для початку, виділіть цю таблицю, потім на закладці «Создание» в групі «Формы» натисніть кнопку «Мастер форм».



Відкриється вікно майстра: перший крок:



Із списку Таблицы и запросы виберіть таблицю Товары (позиція 1 на малюнку).

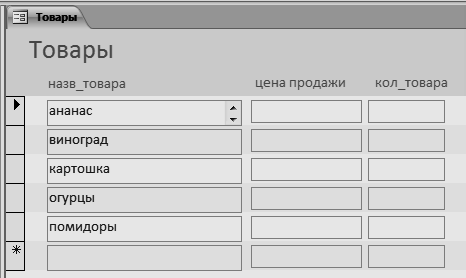
За допомогою кнопок, що позначені на малюнку позицією 2, перенесіть поля з лівого вікна (Доступные поля) у праве вікно (Выбранные поля).

Зверніть увагу: поле Код товара потрібно залишити невибраним!

Нажмить кнопку Далее.

На другому кроці майстра потрібно вибрати режим форми: "Ленточный".

На цьому робота з майстром закінчується, натисніть кнопку Готово і отримаєте результат, як показано на малюнку нижче:



Процес створення форми з майстром, трохи складніший, ніж автоматичне створення, але і результат біль наближений до наших побажань.

Тепер самостійно створіть стрічкові форми для таблиць Продажи і Покупки.

3. Редагування форм в режимі конструктора.

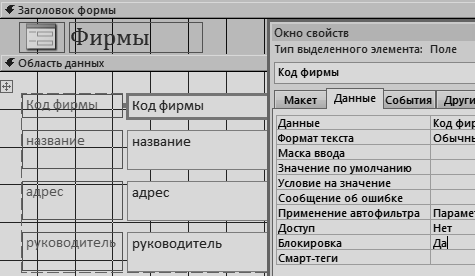
3.1. Завдання: заборонити доступ до ключового поля у формі Фирмы. Згадаємо – форма була створена за один крок. Тому в ній присутні елементи, які насправді нам не потрібні в цій роботі. Тут зараз є поле, яке приєднане до ключового поля Код фирмы. Це поле має тип Счетчик, таке поле не підлягає редагуванню, бо воно автоматично нарощує своє значення з кожним новим рядком у таблиці. Нам потрібно зробити так, щоб користувач бачив цифри коду, але не міг редагувати його.

Відкрийте форму у режимі конструктора. Виділіть поле Код фирмы і викличте його властивості. Викликати вікно властивостей можна через контекстне меню, або через кнопку на панелі інструментів kn_svo

Відкрийте закладку Данные і встановіть:

* Доступ – Нет; Блокировка – Да

Як це виглядає в конструкторі – дивись на малюнку нижче:



Повернувшись у режим форми, Ви побачите, що тепер помістити курсор в поле Код фирмы неможливо, але воно продовжує відображатись на формі.

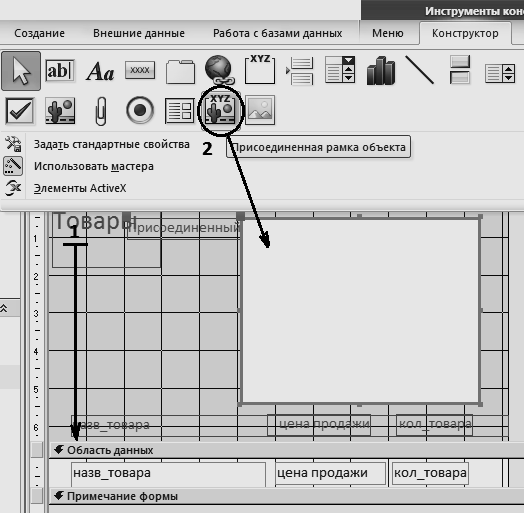
Тепер нам залишилось тільки прибрати зайву підлеглу таблицю Покупки, яку Access помістив до нашої форми. Приєднання було виконано із-за автоматично встановленого зв’язку між таблицями, в процесі створення таблиці Покупки.

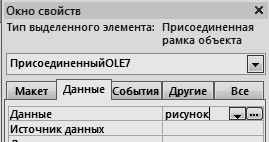
Поверніться в режим конструктора, виділіть таблицю і натисніть клавішу Delete. Завдання виконано. Залишилось тільки зберегти зміни після редагування.

3.2. Завдання: додати у форму Товары можливість відображення товарів у малюнках.

Для можливості зберігання в таблиці додаткової інформації необхідно змінити її структуру. Відкрийте таблицю Товары в конструкторі і добавте нове поле з ім’ям рисунок і типом "Поле объекта OLE". Збережіть зміни та закрийте таблицю.

Відкрийте в конструкторі форму Товары. Збільшіть розмір заголовку форми, для розміщення на ньому малюнку та перемістіть надписи полів донизу області заголовку. На закладці Конструктор розкрийте групу Элементы управления і добавте в область заголовка форми елемент "Присоединенная рамка объекта"

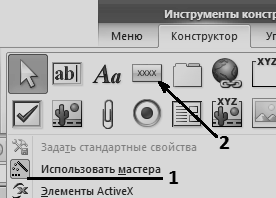


Встановіть для цього елементу властивість Данные - рисунок (ім’я поля). Виконується така дія у вікні властивостей на закладці Данные.

На закладці Макет встановіть властивість Установка размеров – «По размеру рамки»

Збережіть зміни та перейдіть в режим форми. Тепер Ви можете вставляти зображення до ваших товарів. Це можна зробити будь-яким зручним для вас способом. Наприклад, знайдіть малюнок в Організаторі кліпов, помістіть його в буфер обміну, поверніться в Access і вставте в рамку об’єкта малюнок із буфера через контекстне меню на рамці.

3.3. Створення активних елементів на формі – кнопки для виконання дії: «закрити форму».

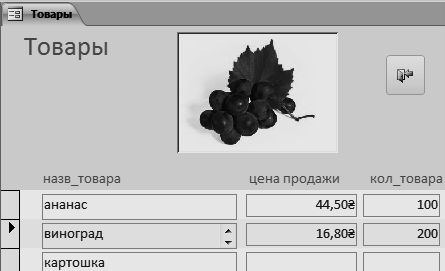


Перейдіть в режим конструктора форми. Перевірте: чи натиснута кнопка «Мастера» на панелі інструментів «Элементами управления». Дивись малюнок позиція 1. За допомогою елементу управління «Кнопка» (поз.2) розмістіть кнопку на формі в області заголовку.

Запуститься майстер:

* 1-й крок майстра – вибір категорію і дію:   
  Категория -"Работа с формой"; Действие - "Закрытие формы". Натисніть кнопку «Далее»;
* 2-й крок – потрібно вибрати малюнок для кнопки, або напис, який буде відображатись на кнопці.
* На 3-му кроці задайте ім’я кнопці. Це ім’я буде використовуватись в програмних модулях, тому його краще задавати англійським алфавітом і у стандартах програмування.

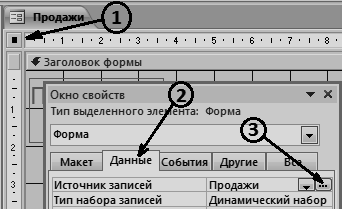
Отримаємо щось на зразок цього:



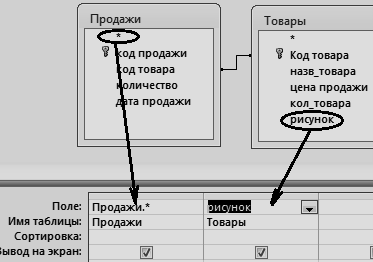
3.4. Редагування форми Продажи. В область заголовку форми потрібно вивести малюнок для вибраного товару. При цьому, малюнок повинен тільки показуватись без можливості редагування.

Поле, що містить малюнок, знаходиться в таблиці Товары. Але в таблиці Продажи ми маємо поле код товара. Для реалізації нашого завдання, потрібно створити запит, який зв’яже ці дві таблиці і який матиме поле рисунок. Далі ми зможемо використати цей запит як джерело записів для форми Продажи.

Відкрийте форму Продажи в конструкторі. Виведіть вікно властивостей форми. Для цього виділіть всю форму в конструкторі (поз.1 на малюнку нижче). На закладці Данные (поз.2) властивість Источник записей містить назву таблиці Продажи. Його потрібно замінити на інструкцію SQL. Цю інструкцію нам допоможе створити Построитель запросов. Для його виклику натисніть кнопочку праворуч властивості Данные. (поз.3).



З’явиться вікно конструктора запиту в якому вже буде одна таблиця - "Продажи". Щоб додати в запит нову таблицю, використовуйте кнопку на панелі інструментів kn_sql1.



Після додавання таблиці Товары, автоматично з’явиться зв’язок між обома таблицями по полям Код товара. Тепер перетягніть мишкою в перше поле запиту зірочку із структури таблиці Товары (як показано на малюнку вище). Це означає, що ми включаємо до запиту всі поля цієї таблиці. В друге поле запиту перетягніть поле рисунок із таблиці Товары. Закрийте конструктор запиту. При закритті, Access запитає: чи вносити зміни? Підтвердіть. Тепер у властивості Источник записей замість назви таблиці з’явиться інструкція SQL:

SELECT Продажи.\*, Товары.рисунок FROM Товары INNER JOIN Продажи ON Товары.[Код товара] = Продажи.[Код товара];

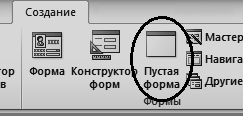
Тепер Вам залишилось самостійно додати в заголовок форми рамку зв'язаного об'єкта і приєднати її до поля рисунок. Виконується так само, як це описано в пункті 3.2. поточної лабораторної роботи.

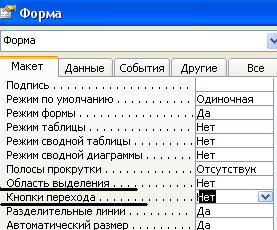
Крім того потрібно обмежити доступ до редагування об’єкту. Виконується ця дія так само, як це описано в пункті 3.1. поточної лабораторної роботи.

Завершення редагування полягає у додаванні до форми активної кнопки для закриття форми. Виконується ця дія так само, як це описано в пункті 3.3. поточної лабораторної роботи.

4. Створення форми з кнопками для запуску різних об’єктів БД.

На закладці Создание група Формы кнопка Пустая форма



Створиться форма без джерела записів. Далі поставте на неї кілька кнопок, що будуть запускати всі ваші наявні в БД форми. Всі відомості про виконання цієї вправи ви вже отримали, виконавши п.3.3. поточної лабораторної роботи.

Додатково потрібно модифікувати зовнішній вигляд форми. Оскільки форма не має джерела записів, на ній не потрібно відображати елементи, що пов’язані з обробкою записів. Змініть такі властивості форми:

# Лабораторна робота № 2.3 MS Access - запити і створення складних форм.

Мета роботи: Поглибити знання про можливості Access, навчитись розробляти складні запити і форми, створювати та редагувати модулі VBA, для автоматизації роботи БД.

### Короткі відомості

#### Запити.

Запит - це текстовий об'єкт MS Access, що містить інструкцію SQL. Мова SQL (Structured Query Language) використовується для управління реляційними базами даних, у тому числі і БД MS Access.

Інструкція SQL – це запис, який починається командою SQL, за якою слідує речення SQL

Команда SQL - це зарезервовані даною мовою ключові слова: SELECT, UPDATE, DELETE і т.д.

Речення SQL складається також із ключових слів (наприклад, FROM, WHERE, ORDER BY) а також назв таблиць і полів.

Інструкції SQL використовуються як запити, і як джерела записів для форм, звітів і списків. MS Access часто автоматично створює інструкцію SQL. Наприклад, при використанні майстра для створення форми або звіту, який отримує дані з декількох таблиць, або при створенні поля зі списком.

#### Типи запитів

Запити за їх призначенням можна умовно розділити на два типи: *Запити для відображення даних* і *Керуючі запити*.

##### Запити для відображення даних

Використовуються для вибірки тих записів, які задовольняють заданим умовам, або для представлення даних в потрібному для нас вигляді. За допомогою цих запитів можна відображати інформацію з декількох таблиць.

Результатом виконання даного типу запитів є набір записів у вигляді таблиці. Такий набір називається Recordset.

До даного типу відносяться *запити на вибірку* та *перехресні запити*.

Запити на вибірку. Ці запити використовуються найчастіше. При його виконанні дані, що задовольняють умовам відбору, вибираються з однієї або декількох таблиць і виводяться в певному порядку. Запити на вибірку створюються за допомогою команди мови SQL SELECT.

В загальному вигляді синтаксис інструкції виглядає так:

SELECT список\_полів\_для\_відображення

FROM імена\_таблиць\_які\_містять\_потрібні\_дані

WHERE вираз\_що\_визначає\_умови\_відбору

ORDER BY і’мя\_поля\_по\_якому\_виконується\_сортування;

Третій і четвертий рядок в цьому записі не є обов'язковими (їх може і не бути). Ознакою закінчення інструкції є знак «крапка з комою».

Для створення запитів в MS Access використовується Конструктор запитів.

Обчислювані поля Такі поля запитів призначені для отримання додаткової інформації за рахунок обробки даних з інших полів. Вони дозволяють уникнути зберігання в БД надлишкової інформації. Обчислювані поля створюються за допомогою виразів. Наприклад: Виторг: [ціна продажу] \* [кількість]

У даному прикладі «Виторг» - назва обчислюваного поля, після нього обов'язково ставиться знак двокрапки. Потім починається вираз. Квадратні дужки – це службові символи, в них заключають імена полів, при написанні виразів.

##### Керуючі запити.

Ці запити призначені для виконання дій над даними. За допомогою них можна змінювати значення в таблицях, додавати і видаляти записи з таблиць. До таких запитів відносяться: *Запит на оновлення, Запит на додавання, Запит на видалення, Перехресний запит*.

##### Події

Важливою особливістю форм є те, що вони можуть містити активні елементи і навіть додаткові модулі, що по своїй суті являються програмними засобами. Таким чином форми дають можливість програмувати процес роботи з БД.

«События» – це події, що можуть відбуватися з об’єктом, в процесі роботи форми. Для кожної з таких подій можна створювати процедури обробки подій. Саме ці процедури являються інструментом для програміста і дозволяють створювати складні додатки в MS Access. Деякі процедури обробки подій створюються автоматично. Наприклад, якщо майстром встановлювати кнопку на форму для дії «закрити форму», то для кнопки буде автоматично створена процедура з ім'ям <імя\_кнопкі> \_Click (). Для перегляду коду процедури треба відкрити вікно властивостей, перейти на вкладку «События» і викликати редактор VBA.

### Робоче завдання:

1. Створення обчислюваного поля в формі та відображення в поточній формі даних з інших таблиць.
2. Реалізація автоматичної перевірки даних, що вводяться в форму.
3. Реалізація автоматичної обробки даних із інших таблиць, в залежності від даних, що вводяться в поточну форму.
4. Створити активні елементи для сортування даних за бажанням користувача.
5. Самостійна розробка сервісних елементів форм.

### Порядок виконання завдання

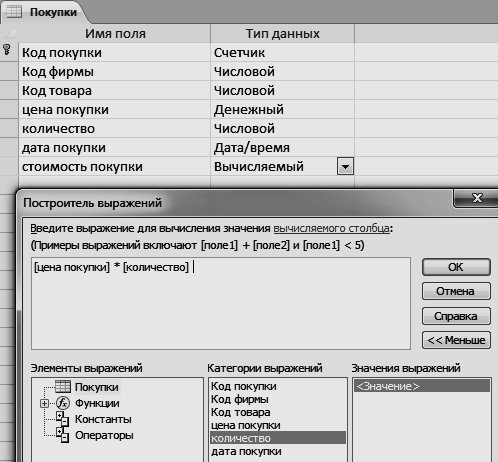
Ми будемо з вами змінювати форму Покупки, таким чином, щоб вона стала дійсно зручною для роботи оператора при масовій обробці даних. Ця форма була створена Вами в попередній роботі за допомогою майстра (див. п.2 лабораторної роботи №2.2)

1. Створення обчислюваного поля.

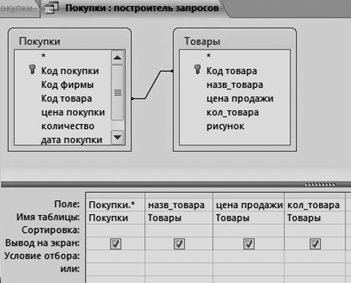
При купівлі товару треба автоматично формувати загальну вартість закупівлі, за виразом: [количество]\*[цена покупки].

В нашій вправі, для виконання такої задачі, ми змінимо структуру таблиці Покупки. Відкрийте таблицю в режимі конструктора і створіть нове поле стоимость покупки з типом Вычисляемый.

Відкриється вікно для побудови виразів. Створіть у ньому вираз, як показано на малюнку нижче.



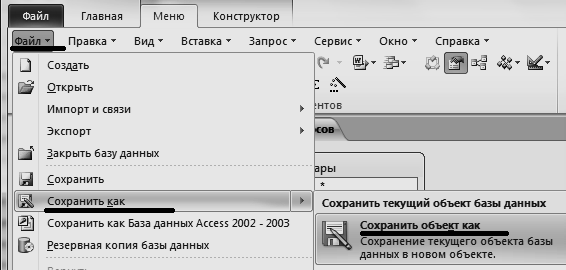
Крім того, нам потрібно відобразити на поточній формі Покупки інформацію із таблиці Товары, для запобігання помилок оператором. Це потребує зміни такої властивості форми, як *Источник данных*. Поки що ця властивість містить назву таблиці **Покупки**. Замість неї нам потрібно поставити запит:



Включіть у запит: всі поля із таблиці Покупки та поля назв\_товара, цена продажи, кол товара із таблиці Товары.

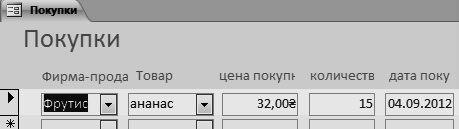
Збережіть створений запит під ім’ям Закупки. Такими чином ми створимо новий об’єкт БД і він тепер буде використовуватись у якості джерела даних для форми Покупки.

Для цього Вам потрібно виконати команду Сохранить объект как із меню Файл (деталі дивіться на малюнку нижче). З’явиться діалогове вікно, де потрібно вказати ім’я для нового об’єкту – вкажіть ім’я для нового запиту: Закупки.

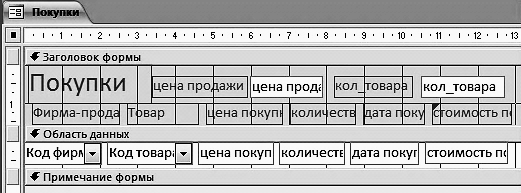


Закрийте конструктор запиту. З’явиться питання про збереження внесених змін – підтвердіть свої дії.

Тепер будемо змінювати саму форму. Ось що у нас зробив майстер в попередній роботі:



Для внесення додаткових даних (які з’явились при створенні нового джерела даних для форми) ми розширюємо заголовок форми і додаємо туди інформацію з таблиці Товары. А в область даних додаємо поле стоимость покупки, яке ми створили на початку цієї роботи.



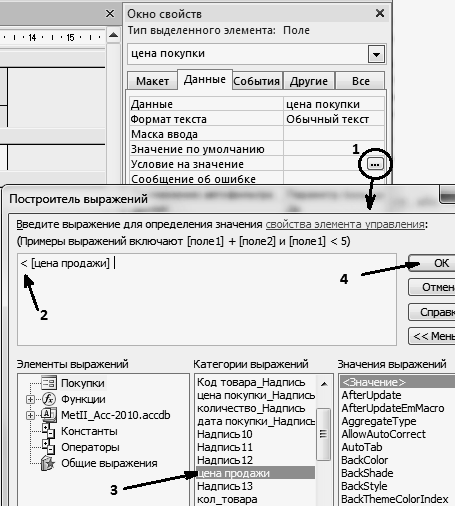
Закрийте доступ оператору до всіх полів, що знаходяться в заголовку форми (див. п.3.1. попередньої лабораторної роботи).

2 Реалізація автоматичної перевірки даних, що вводяться в форму. Організуємо перевірку за ціною закупівлі. Вона не повинна перевищувати ціну, за якою ми продаємо товар. Для цього у властивість Условие на значение поля цена покупки встановіть наступний вираз:

< [цена продажи]

(менше ціні з таблиці Товары).

Цей вираз можна написати вручну, або створити за допомогою *Построителя выражений*:

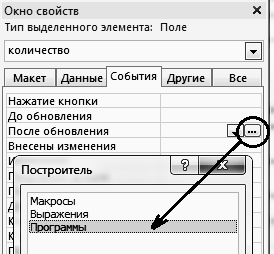


Тепер, якщо оператор буде вводити в поле ціну, що дорівнює, або перевищує ціну продажу (що зберігається в таблиці Товары), то Access буде видавати стандартне повідомлення про помилку. Ми, як розробники форми, маємо змогу замінити стандартне повідомлення своїм, більш зрозумілим оператору. У значення властивості Сообщение об ошибке запишіть свій текст. Наприклад: "*При такій закупівлі ми понесемо ЗБИТКИ! Встановіть меншу закупівельну ціну*."

3. Реалізація автоматичної обробки даних із інших таблиць, в залежності від даних, що вводяться в поточну форму.

**Завдання**: потрібно автоматично змінювати кількість товару, що зберігається в таблиці Товары в полі кол товара, зразу ж після введення даних про кількість закупленого товару у формі Покупки.

Тут ми будемо використовувати групу властивостей События (Події). У поля количество покупки є властивість После обновления. Саме для цієї події ми створимо процедуру обробки.



Після натискання кн. ОК, відкриється редактор коду:

Private Sub количество\_AfterUpdate()

End Sub

Ви побачите таку процедуру обробки. Тут вже є початок та кінець процедури. Немає самого коду. Його будемо писати самі.

Private Sub количество\_AfterUpdate()

[кол\_товара] = Nz([кол\_товара], 0) +[количество].Value - Nz([количество].OldValue, 0)

End Sub

Після внесення таких змін у форму, вона буде виконувати автоматичну обробку даних.

Пояснення до коду:

Найпростіший варіант: це просто додати нову кількість товару до тієї, що зберігається в таблиці Товары. Але такий варіант не буде враховувати можливість виправлення помилки у існуючому записі.

У загальному випадку, для підрахунку нового значення загальної кількості товару, можна застосувати формулу:

New\_k = Old\_k + n\_kol\_p - s\_kol\_p

де:

New\_k - нова кількість товару (для збереження в таб.Товары)

Old\_k - кількість товару, записана в таб.Товары до редагування запису у формі Покупки.

n\_kol\_p - нове значення в полі количество покупки (по закінченню редагування)

s\_kol\_p - старе значення поля количество покупки

Нижче наведено **чорновий варіант коду** процедури, де видно всі кроки підрахунку:

Private Sub количество\_AfterUpdate()

Dim New\_k, Old\_k, s\_kol\_p, n\_kol\_p

Old\_k = [кол товара]

s\_kol\_p = [количество].OldValue

n\_kol\_p = [количество].Value

New\_k = Old\_k + n\_kol\_p - s\_kol\_p

[кол товара] = New\_k

End Sub

* 1-й рядок: оголошення функції;
* 2-й рядок: визначення змінних, що будуть використовуватись далі;
* 3-й рядок: отримуємо в змінну кількість товару, яка є на поточний момент;
* 4-й рядок: зчитуємо у змінну властивість поля OldValue (значення до редагування поля);
* 5-й рядок: зчитуємо у змінну властивість поля Value (значення після редагування поля);
* 6-й рядок: підрахунок нової кількості товару
* 7-й рядок: заносимо підраховане значення у поле
* 8-й рядок: закінчення функції

Такий текс коду приведено для детального показу процесу обробки. **Спрощений варіант коду** отримаємо, якщо не використовувати змінні, а одразу провести підрахунок, взявши значення із полів:

Private Sub количество\_AfterUpdate()

[кол товара] = [кол товара] + [количество].Value - [количество].OldValue

End Sub

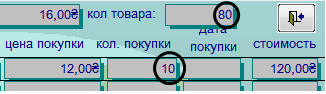
Нажаль, така процедура не враховує ще деякі особливості підрахунку. А саме: поля можуть бути з невизначеним значення. Тобто, до того моменту, доки в поле не занесене якесь значення, поле в Access визначається як Null (значення не присвоєно). Математичні операції з таким полем виконувати не можна – це приведе до виникнення помилки. Для уникнення помилки, потрібно замінити можливе значення Null на арифметичний нуль. В Access таку операцію найпростіше виконати за допомогою функції Nz(). Синтаксис функції:

Nz(об’єкт, значення\_якщо\_об’єкт\_Null)

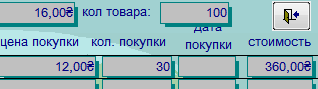
Ця функція перевіряє об’єкт на значення Null і, якщо воно НЕ Null, то повертає сам об’єкт, а якщо Null, то повертає свій другий параметр.

Таким чином: Nz([кол товара].OldValue, 0) поверне число, що внесене в це поле, або нуль, якщо поле не мало значення до початку редагування (така ситуація виникне при внесенні нового запису).

На малюнку зображено стан запису до початку редагування:



А ось після зміни кількості із 10 на 30:



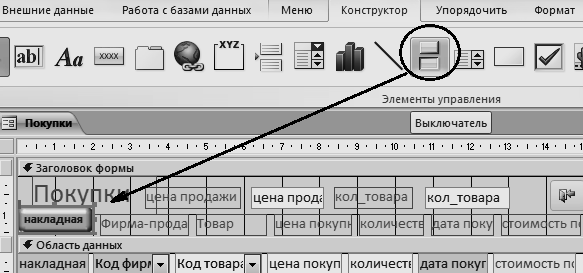
Поля кол товара і стоимость змінились автоматично, без участі оператора. Зверніть увагу: ми з вами внесли зміни в таблицю Товары, хоч працювали з формою Покупки.

5. Створення активних елементів для сортування даних.

Для цього використаємо елемент управління "выключатель". Такий елемент приймає різні значення, в залежності від свого стану. Якщо вимикач натиснуто (увімкнено) – True і навпаки, якщо вимкнено – False.

Ці значення можна зчитувати і змінювати стан сортування форми в залежності від дій користувача. Крім того, ці значення можна і присвоювати елементу управління (в процесі виконання коду процедури обробки подій), тим самим примусово переводячи елемент управління в потрібний стан.

Створіть в таблиці Покупки поле Накладная, додайте його до форми Покупки та заповніть. Замість назви поля накладная поставте вимикач.



Задайте наступні властивості цьому елементу управління:

* властивість Имя = sort\_nom   
  (закладка Другие)
* властивість Подпись = накладная  
   (закладка Макет)

На місці інших написів у заголовку створіть ще кілька вимикачів, дайте їм відповідно імена та підписи. Тепер нам потрібно написати процедури обробки подій. Наприклад:

Private Sub sort\_firm\_Click()

If Me!sort\_firm Then

Me.OrderBy = "[Код фирмы]"

Me.OrderByOn = True

Else

Me.OrderByOn = False

End If

Me!sort\_nom = False

Me!sort\_date = False

- - - - - - -

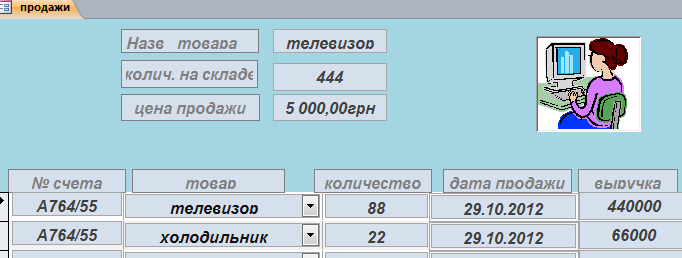
End Sub

Пояснення до коду:

* 1-й рядок: оголошення функції;
* 2-й рядок: початок блоку перевірки стану вимикача;
* 3-й і 4-й рядки виконуються, якщо вимикач знаходиться у натиснутому стані. 3-й –властивість OrderBy (сортувати по….) об’єкту форма змінюємо на текст, який містить назву потрібного поля (для кожної кнопки буде своє ім’я поля). 4-й рядок коду – активуємо сортування;
* 5-й рядок: оператор, за яким починається друга частина блоку перевірки;
* 6-й рядок виконується, якщо кнопку переведено у стан «вимкнено». В такому випадку потрібно зняти сортування з форми, що и виконується деактивацією властивості сортування форми;
* 7-й рядок: оператор закінчення блоку перевірки;
* 8-й, 9-й (і можуть бути ще рядки) – переводимо всі інші існуючі перемикачі у стан «вимкнено»;
* кінець функції.

6.Самостійно змініть форму Продажи як показано на малюнку. Звичайно, при продажі, кількість відповідного товару на складі (таблиця Товары), повинна зменшуватись**.**

Заповніть форми Покупки і Продажи не менше 15 записами. так, щоб у фірм на одну накладну було закуплено різний товар. По цих записах ми будемо створювати звіти.

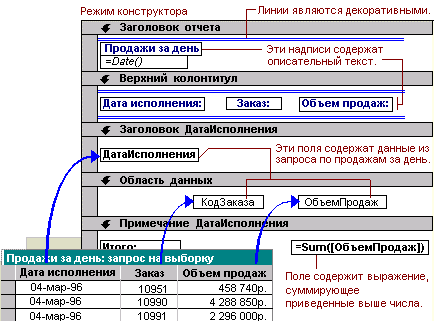


# Лабораторна робота № 2.4 MS Access - Звіти, Кнопкові форми та параметри запуску БД

Мета роботи: Отримати знання і навики по створенню звітних документів із існуючої БД. Навчитись розробляти спеціальні багаторівневі Кнопкові форми, за допомогою спеціальної надбудови MS Access. Ознайомитись з можливостями різних варіантів запуску готової БД.

### Короткі відомості

Звіт - це гнучкий і ефективний засіб для організації даних при виведенні на друк. За допомогою звіту є можливість вивести необхідні дані в тому вигляді, в якому потрібно.



Основні дані для звіту беруться з базової таблиці, запиту або інструкції SQL, вони є джерелом даних для звіту. Для відображення даних із джерела даних застосовуються елементи управління (так само, як у формах) – поля, списки. Також у звітах можна використовувати поля, що обчислюються в процесі формування звіту. Крім того у звітах можуть бути використані елементи управління для оздоблення – написи, лінії або інші графічні об'єкти.

Особливістю звітів є групування даних. Створюючи звіт можна зібрати всі записи з однаковим значенням і надрукувати їх як окремий блок звіту. Такий блок називається групою. Група має заголовок, область даних і примітки групи. У заголовку, як правило, виводиться значення того поля, по якому проведено угруповання. В область даних включаються ті поля, по яких не проводилося угруповання. А в примітки вставляються різного роду підсумки: кількість записів, сума по стовпцях і т.і. Групувати дані в MS Access 2010 можливо в декілька рівнів (до 4 рівнів максимально).

Наприклад, якщо у нас є база студентів всього університету, то її можна згрупувати за наступними рівнями:

* 1-й рівень: назва факультету;
* 2-й – курс (або рік вступу);
* 3-й – назва групи

У такому прикладі, в область даних можна вивести інформацію про студента: ПІБ; № залікової книжки; т.і. Так ми отримаємо списки студентів структуровані по факультетам, курсам і групам. В примітки кожної групи можна включити підсумкові поля з підрахунками, наприклад: загальна кількість студентів; контрактників тощо.

У зв’язку з такою особливістю, звіти (на відмінність від форм) мають дещо більший набір структурних елементів:

* Заголовок звіту;
  + Верхній колонтитул
    - Заголовок групи-1
      * - - - - - - - - - - - - -
        + Заголовок групи-4

Область даних .

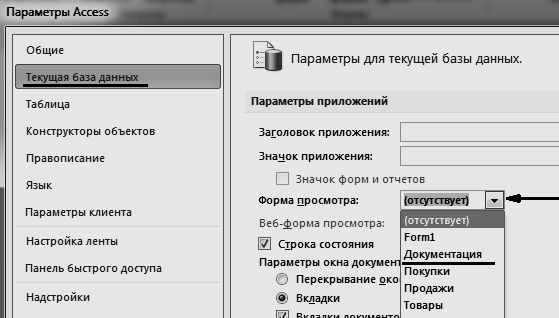
* + - * + Примітки групи-4
      * - - - - - - - - - - - - -
    - Примітки групи-1
  + Нижній колонтитул
* Примітки звіту.

#### Форма навігації

Кожна біль-менш складна БД містить досить великий набір різних об'єктів. Для їх запуску можна використовувати область навігації БД. У програмі Access 2010 доступний новий елемент навігації, який спрощує перехід між різними формами та звітами в базі даних. Форма навігації являє собою просту форму, що містить елемент навігації. Форми навігації добре доповнюють будь-яку базу даних, але особливо важливо створити їх, якщо база даних буде опублікована в Інтернеті, оскільки область навігації Access не відображається в браузері.

#### Параметри БД

Відповідне діалогове вікно викликається через вкладку стрічки   
Файл ->Справка ->Параметры:



Його доцільно використати на завершальній стадії розробки БД. Тут можна встановити текс, що буде відображатись в заголовку вікна Вашого застосування. Також можна вибрати індивідуальну іконку замість стандартного зображення.

Крім того, MS Access дає можливість керування запуском бази даних. Розробник БД може вказати, яку форму слід відкрити при запуску, чи дозволяється використовувати панелі інструментів, чи можна їх змінювати, а також чи є доступними: вікно БД; рядок статусу; стандартні контекстні меню.

Всі ці налаштування, в основному, пов’язані із обмеженнями для користувача в частині внесення змін до роботи БД. Тобто примушують користувача чітко виконувати тільки ті дії, що дозволив для нього розробник БД.

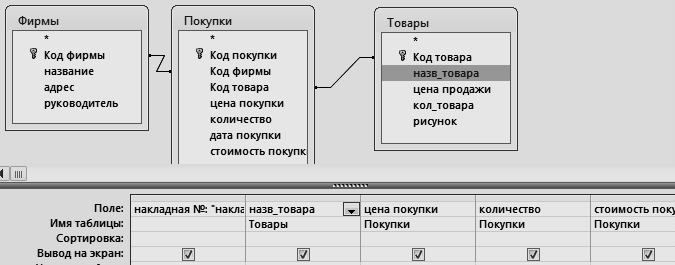
### Робоче завдання:

1. Створення звітів за допомогою майстра. Редагування звітів в конструкторі..
2. Створити форму навігації для зручної роботи з БД.
3. Встановити особливі параметри БД.

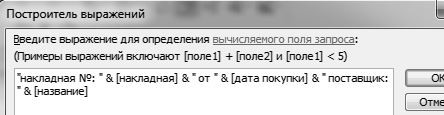
### Порядок виконання завдання

1. Створення звіту. Звіт можна створити декількома способами. Всі вони відображені на вкладці «Создание» група «Отчеты».

Для звіту потрібне джерело даних. У нашому прикладі створимо новий запит Покупки для отчета. Цей запит буде базуватись на таблицях Покупки, Товары, Фирмы а містити він буде одне збірне поле і декілька полів для звіту по закупкам.



Перше поле в цьому запиті потрібно зробити через вікно будівника виразів:



В полі виконується операція «конкатенація» (об’єднання) текстових рядків і значень полів. Результатом такого об’єднання стане новій текстовий рядок.

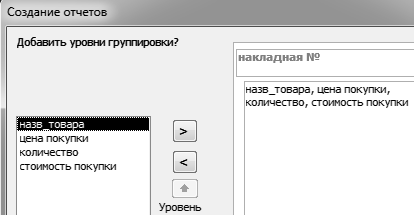
SQL цього запиту буде виглядати так:

SELECT "накладная №: " & [накладная] & " от " & [дата покупки] & " поставщик: " & [название] AS [накладная №], Товары.назв\_товара, Покупки.[цена покупки], Покупки.количество, Покупки.[стоимость покупки]

FROM Фирмы INNER JOIN (Товары INNER JOIN Покупки ON Товары.[Код товара] = Покупки.[Код товара]) ON Фирмы.[Код фирмы] = Покупки.[Код фирмы];

Тепер виділіть створений запит і викличте майстра звітів. Далі почне працювати майстер звіту. *На першому кроці* потрібно відібрати ті поля, що будуть використовуватись у звіті. Потрібно вибрати всі доступні поля, адже ми створювали запит саме для цього.

*На другому кроці* створюються рівні групування. Зробіть групування по полю накладная №:



*На третьому кроці* майстра, можна вказати порядок сортування даних. Встановіть сортування по полю назв\_товара. Крім того, тут можна встановити підсумки у звіті.

*На четвертому кроці* дає можливість вплинути на зовнішній вигляд звіту. Як правило, звіт все одно буде потрібно доопрацьовувати в конструкторі, тому можете обрати будь який із запропонованих варіантів.

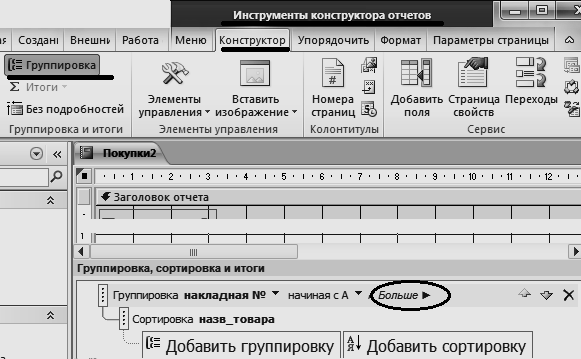
Останній *п’ятий крок* – задати назву звіту. І отримаємо:



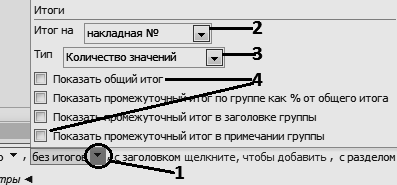
Далі відредагуйте отриманий звіт у конструкторі, для найкращого представлення даних. У Вашому завданні потрібно:

* відкоригувати назви стовпчиків;
* створити примітки групи, куди включити загальну вартість закупівлі і загальну кількість записів по групі.

Для включення приміток групи виконайте наступні дії. Стрічка «Инструменты конструктора отчетов» закладка «Конструктор» кнопка «Группировка» - в нижній частині конструктора з’явиться додатковий розділ: «Группировка, сортировка и итоги». Натисніть область «Больше». На малюнку нижче це показано наглядно:



Відкриється ряд додаткових параметрів групування. Спочатку там стоятиме надпис «без итогов» натисніть біля нього кнопку (1). З’явиться віконце:

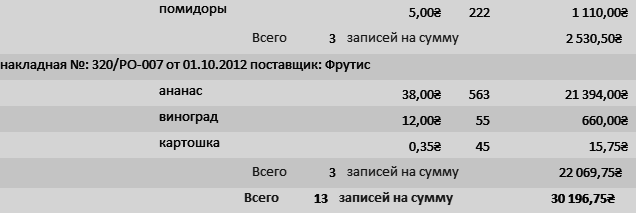


(2) Виберіть в списку Итог на: накладная №;

(3) в списку Тип – Количество значений. Після того, як ви включите два параметри, що зображені на малюнку під №4, у звіті автоматично з’являться два поля в примітках групи і примітках звіту. Це будуть вільні елементи з підрахунком кількості записів.

Далі в списку (2) виберіть назву поля стоимость покупки; а в списку (3) – Сумма; увімкніть поля, позначені як (4). Створиться ще два поля для підрахунку вартості закупівлі.

Самостійно змініть форматування цих полів, для того, щоб звіт став виглядати приблизно так:



Далі самостійно створіть ще 2 звіти:

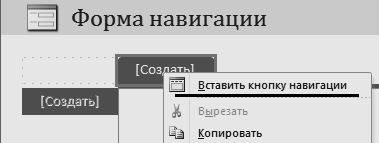
* по продажам із групуванням по назві товару і сортуванням по даті продажу та з підсумками по сумі реалізації.
* по закупкам із групуванням по даті закупівлі (1-й рівень) і назві товару (2-й рівень), сортування і підсумки придумайте самі.

2. Форма навігації.

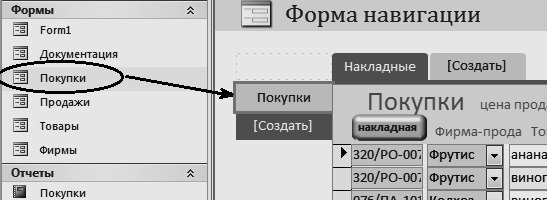
Вам потрібно створити форму із дворівневою структурою, що буде виконувати наступні дії:

1. Внесення поточних даних до БД – горизонтальна закладка Накладные
   1. Вертикальна закладка – форма Покупки
   2. Вертикальна закладка – форма Продажи
2. Друк звітів – горизонтальна закладка Отчёты
   1. Вертикальна закладка – звіт Закупки
   2. Вертикальна закладка – другий звіт
   3. Вертикальна закладка – третій звіт
3. Редагування довідників – горизонтальна закладка Справочники
   1. Вертикальна закладка – форма Товары
   2. Вертикальна закладка – форма Фирмы

**Зверніть увагу**: перший рівень наведеної структури реалізується як горизонтальні вкладки навігації, а другий рівень – як вертикальні вкладки. Тобто Вам потрібно при створенні форми навігації вибрати відповідний тип форми навігації. Виконується: стрічка Создание, група Формы, кнопка Навигация. Відкриється вікно нової форми в режимі макету. На верхній кнопці викличте контекстне меню і виберіть команду Вставить кнопку навигации.

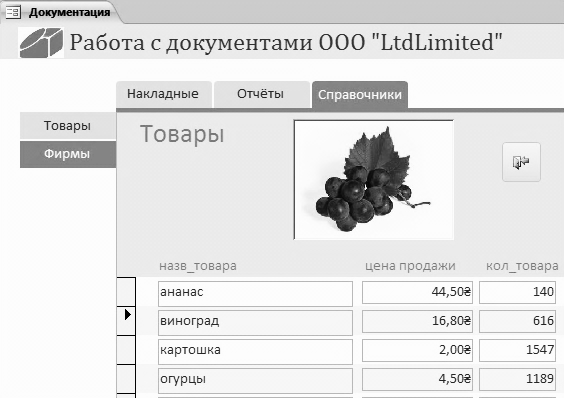


Далі клацніть двічі по створеній кнопці і напишіть там текст: «Накладные». Тепер перетягніть мишкою із області навігації БД форму Покупки на місце вертикальної кнопки



На другу кнопку перетягніть форму Продажи.

Далі самостійно створіть всі інші кнопки із наведеної вище структури. Змініть заголовок і підпис форми. Приклад нижче:



3. Параметри БД. Після закінчення роботи по розробці всіх компонентів і об’єктів Вашої бази даних залишається тільки налаштувати параметри, з якими БД буде запускатись.

Вам потрібно:

1. Зробити унікальний заголовок для вашого застосування – придумайте самі.
2. Знайдіть будь який файл \*.ico і встановіть його у якості значка Вашого застосування.
3. Встановіть в якості стартової форму навігації. Тобто вона буде автоматично відкриватись при запуску БД.
4. Сховайте від користувача область навігації.

# Контрольні питання:

1. СУБД Microsoft Access. Створення БД.
2. Головне вікно MС Access. Поняття файла-контейнера.
3. Проектування таблиць, типи даних полів, встановлення параметрів поля.
4. Редагування таблиць, механізм підстановки.
5. Поняття ключового поля та індексу. Типи міжтабличних зв’язків.
6. Створення та призначення об’екту Форма. Діалогове вікно «Свойства».
7. Режим конструктора форми. Призначення Панелі елементів.
8. Створення запитів. Робота в конструкторі запитів
9. Типи запитів, їх призначення.
10. Створення звітів. Групування записів, структура групування.
11. Робота в конструкторі звітів.
12. Діалогове вікно «Параметры запуска.»
13. Експорт-імпорт даних. Використання зовнішніх зв’язаних таблиць.

# Література

1. Т.В.Тимошок Microsoft Access 2003 Навч. посіб. –М.Диалектика, 2005. –315 с.
2. Дибкова Л.М.. Інформатика та комп'ютерна техніка: Посібник. – К: Вид. центр "Академія", 2002. – 320 с.
3. Карпенко С.Г., Іванов Є.О. Основи інформаційних систем і технологій: Навч. посіб. – К.:МАУП, 2002. – 264с.
4. Інформатика: Комп'ютерна техніка. Комп'ютерні технології: Підручник для студентів вищих навчальних закладів / За ред. О.Пушкаря. – К: Видавничий центр "Академія", 2002. – 704 с
5. Рзаєв Д.О., Шарапов О.Д., Ігнатенко В.М., Дибкова Л.М. Інформатика та комп'ютерна техніка: Навч. –метод. пос. –К.: КНЕУ, 2002. –486 с.
6. Ярмуш О.В., Редько М.М. Інформатика та комп'ютерна техніка:Навч.посібник. –К.: Вища освіта, 2006. –359 с.